



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ Offenlegungsschrift
⑩ DE 197 35 278 A 1

⑤1 Int. Cl.⁶:
G 06 F 11/00

②1 Aktenzeichen: 197 35 278.2
②2 Anmeldetag: 14. 8. 97
④3 Offenlegungstag: 18. 2. 99

DE 197 35 278 A 1

⑦1 Anmelder:
Wadewitz, Rolf, 59755 Arnsberg, DE

⑦4 Vertreter:
Mitscherlich & Partner, Patent- und Rechtsanwälte,
80331 München

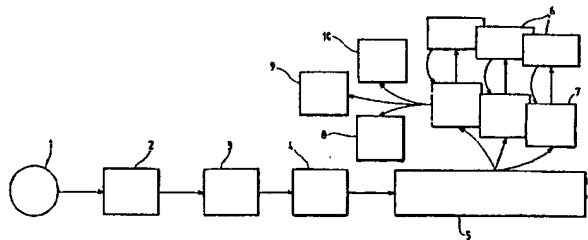
⑦2 Erfinder:
gleich Anmelder

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Elektronisches Datenerfassungsverfahren und Datenverarbeitungssystem

⑤7 Um zu verhindern, daß Applikationen (6) eines Rechnersystems mit fehlerbehafteten Daten arbeiten, werden alle **Eingaben** vor ihrer Weiterleitung an eine oder mehrere Applikationen (6) auf ihre Richtigkeit überprüft und dabei fehlerbehaftete Eingaben zurückgewiesen oder korrigiert. Diese Überprüfung der **Eingaben** erfolgt durch einen oder mehrere in den Ereignisstrom des Rechnersystems eingebundene Preprozessoren (7). Das Datenerfassungsverfahren ist sowohl bei Singletasking- oder konsolenorientierten Betriebssystemen als auch bei Multitasking- oder fensterorientierten Betriebssystemen einsetzbar.



DE 197 35 278 A 1

Die Erfindung betrifft ein elektronisches Datenerfassungsverfahren nach dem Oberbegriff von Patentanspruch 1, sowie eine Vorrichtung zur Durchführung des Datenerfassungsverfahrens nach Patentanspruch 9.

In der elektronischen Datenverarbeitung (EDV) werden im wesentlichen Daten verarbeitet, gespeichert, abgerufen und verändert. Dabei ist in allen Bereichen der EDV ein Kriterium besonders wichtig, nämlich die Authentizität, Fehlerfreiheit und Zuverlässigkeit der Daten. Selbst das innovativste EDV-System kann nur unbefriedigende Leistungen abliefern, wenn die ihm zugrundeliegenden Daten fehlerbehaftet sind.

Zur Lösung dieser Problematik werden in bekannten EDV-Systemen verschiedene Verfahren angewandt, um einen informationsverlustfreien Transport und eine Authentizitätskennung von Daten zu ermöglichen. Solche Verfahren sind zum Beispiel auf Maschinenbasis, durch Protokolle mit Checksummenberechnung, durch redundante Codes mit der Möglichkeit der Fehlererkennung und -beseitigung oder durch maschinenlesbare Dokumente, wie scanbare Dokumente, Barcodes, Schrifterkennung, OCR, usw., verwirklicht. Bei einer manuellen Dateneingabe können beispielsweise Artikelnummern mit Prüfziffern versehen werden, um eine Fehleingabe zu erkennen und abzuweisen. Alle diese Verfahren haben die Aufgabe, die Konsistenz und Fehlerfreiheit von Daten bei ihrer Übertragung von A nach B zu optimieren und den Informationsverlust bei dieser Übertragung zu minimieren oder zu verhindern.

Eine Hauptquelle für fehlerhafte Daten in EDV-Systemen liegt jedoch im Bereich der Datenerfassung, d. h. wenn die Daten das erste Mal in EDV-lesbarer Form erzeugt werden, wobei Fehler vor allem bei der manuellen Datenerfassung auftreten, bei der ein Benutzer die Daten manuell, in der Regel über ein Terminal, in ein Rechnersystem eingibt. Aufgrund von möglicherweise fehlerhaften Eingaben besteht hier eine sehr hohe Wahrscheinlichkeit für Informationsfehler, welche zudem über die gesamte Lebensdauer der Daten im System bestehen bleiben. Fehlerbehaftete Daten können zum Beispiel falsch geschriebene Namen und Adressen, falsche Artikelnummern und Preisangaben, fehlerhafte Produktionszahlen oder Meßwerte und dergleichen sein. In kritischen Fällen kann beispielsweise auch eine falsche Diagnose beim Arzt, eine zu hohe Dosierung von Medikamenten, ein falsches Ersatzteil bei einer Kfz-Reparatur oder eine zu hohe Reaktortemperatur bei chemischen Prozessen vorliegen.

Die oben beispielhaft genannten Fehleingaben sollten möglichst bereits an der Quelle, d. h. bei der Datenerfassung, vermieden oder korrigiert werden. Es sind verschiedene einfache Verfahren für Applikationen bekannt, um die grundlegenden Elemente der Datenerfassung sicherzustellen. Dies sind zum Beispiel Textfilter oder numerische Filter, logische Tests der Eingaben, Tests auf Längenüberschreitung der Eingabe und dergleichen. Hierdurch kann beispielsweise vermieden werden, daß ein Benutzer in einem Eingabefeld für Telefonnummern auch Vornamen oder in einem Datumsfeld auch den 32. Tag eines Monats eingeben kann. Solche Verfahren zur Eingangsprüfung von Daten werden allerdings häufig in den entsprechenden Applikationen nicht eingesetzt.

Ausgehend von dem genannten Stand der Technik ist es eine Aufgabe der vorliegenden Erfindung, ein Datenerfassungsverfahren und ein EDV-System bereitzustellen, welche gewährleisten, daß die Applikation(en) des Systems nicht mit fehlerbehafteten Eingabedaten arbeiten, und zwar insbesondere auch dann, wenn die Applikation(en) des Sy-

stems nicht selbst mit einer Eingangsprüfung der Daten ausgestattet sind.

Diese Aufgabe wird durch ein Datenerfassungsverfahren mit den Merkmalen des Patentanspruchs 1 bzw. durch ein EDV-System mit den Merkmalen des Patentanspruchs 10 gelöst.

Erfindungsgemäß werden die Eingaben vor ihrer Weiterleitung an eine oder mehrere Applikationen des Rechnersystems auf ihre Richtigkeit überprüft- und dabei fehlerbehaftete Eingaben zurückgewiesen oder korrigiert, so daß die Applikation(en) mit fehlerfreien Daten arbeiten kann (können), auch wenn die Applikation(en) nicht selbst über eine Daten-Eingangsprüfung verfügt(en). Die Überprüfung der Eingaben erfolgt durch einen oder mehrere in den Ereignisstrom eingebundene Preprozessoren.

Ein weiterer Vorteil der Erfindung liegt darin, daß das Datenerfassungsverfahren bzw., das EDV-System sowohl bei Singletasking- oder konsolenorientierten Betriebssystemen, wie DOS, als auch bei Multitasking- oder fensterorientierten Betriebssystemen, wie Windows, einsetzbar ist. Je nach Art des Betriebssystems ist die Überprüfung der Eingaben an unterschiedlichen Stellen in den Ereignisstrom des Rechnersystems eingebunden.

Gemäß einer Weiterbildung der vorliegenden Erfindung werden vor der eigentlichen Nutzung bzw. dem Verwendungszweck des Preprozessors bzw. des Datenerfassungsverfahrens alle relevanten Eingaben automatisch und/oder manuell von dem Preprozessor analysiert. Anhand dieser Analyse erstellt der Preprozessor ein Analyse- und Aktionsprofil, mit dem dann die Überprüfung und Verarbeitung der Eingaben erfolgt.

Vorteilhafterweise kann für jeden Benutzer einer Applikation ein eigenes benutzerspezifisches Analyse- und Aktionsprofil erstellt werden.

Ferner können die Applikationen mit Hilfe des Preprozessors um vom Benutzer erwünschte Funktionen ergänzt werden, die dann als Bestandteil der Applikationen benutzt werden können.

Weitere vorteilhafte Ausgestaltungen und Weiterbildungen der Erfindung sind Gegenstand weiterer Unteransprüche.

Die Erfindung wird im folgenden anhand von bevorzugten Ausführungsbeispielen mit Hilfe der beiliegenden Zeichnung näher erläutert. Darin zeigen:

Fig. 1 eine vereinfachte Darstellung des Ereignisstroms in einem Singletasking-Betriebssystem; und

Fig. 2 eine vereinfachte Darstellung des Ereignisstroms in einem Multitasking-Betriebssystem.

Im wesentlichen kommen derzeit zwei Rechnerstrukturen und Betriebssystem-Arten in Betracht, nämlich konsolenorientierte und fensterorientierte Betriebssysteme. Die konsolenorientierten Systeme sind häufig nur Singletasking-Betriebssysteme, während fensterorientierte Systeme in der Regel Multitasking-Betriebssysteme sind. Im folgenden werden stellvertretend für alle möglichen Rechner-Betriebssysteme anhand von Fig. 1 ein konsolenorientiertes Singletasking-Betriebssystem, wie DOS, und anhand von Fig. 2 ein fensterorientiertes Multitasking-Betriebssystem, wie Windows, beschrieben. Es soll dabei vor allem gezeigt werden, an welcher Stelle ein Benutzer eine Eingabe tätigt und an welcher Stelle die entsprechende Applikation diese Eingabe zur weiteren Bearbeitung erhält. Für einen Fachmann auf diesem Gebiet der Computertechnik sollte es ein leichtes sein, die Erfindung in entsprechender Weise in allen möglichen Rechnersystemen einzusetzen.

Eingaben 1 durch einen Benutzer, im folgenden allgemein auch Ereignisse genannt, können zum Beispiel das Drücken einer Taste, das bewegen der Maus, die Eingabe

mit speziellen Eingabemedien, wie Barcodeleser, Touchscreen, Digitizer, Datenhandschuh und dergleichen, oder auch durch andere Maschinen oder Rechnersysteme erzeugte Eingaben, wie Betriebssysteminterne Nachrichten, Signale auf Bussystemen und Netzwerken, Signale von seriellen Schnittstellen, Signale von Störungsleitungen von CNC-Bearbeitungsmaschinen und dergleichen, sein.

Fig. 1 zeigt zunächst in vereinfachter Darstellung den Ereignisstrom in einer EDV-Anlage mit einem Singletasking-Betriebssystem, wie DOS. Das externe Ereignis, also die Benutzereingabe 1, erzeugt durch die Hardware 2 des Rechnersystems einen internen Interrupt im Rechnersystem, der vom Interrupt-Manager 3 des Rechnersystems verwaltet und weitergeleitet wird. Schließlich wird der interne Interrupt von einem Interrupt-Handler 4, welcher der Hardwarekomponente 2 der Benutzereingabe 1 entspricht, in ein internes Ereignis, d. h. in elektronische Signale, umgewandelt. Dieses interne Ereignis wird bei den bisher bekannten Rechnersystemen dem Betriebssystem 5 des Rechnersystems zugeführt und von diesem an die entsprechende Applikation 6 weitergeleitet, wo es entsprechend verarbeitet wird.

Bei dem erfindungsgemäßen System ist vor das Betriebssystem 5 ein Preprozessor 7 geschaltet. Dieser Preprozessor 7 soll eine Vorbehandlung aller auftretenden Ereignisse durchführen, bevor diese in evtl. veränderter Form als modifizierte Ereignisse an die eigentliche Applikation 6 weitergeleitet werden. Je nach Anwendungsfall kann der Preprozessor 7 auch neue Ereignisse erzeugen, beispielsweise Dialogabläufe steuern, Dialoge mit dem Benutzer steuern, wie Paßwortabfrage zur Freischaltung. Er kann Ereignisse unterdrücken und auch zurückweisen. Dabei kann der Preprozessor 7 zusätzlich mit externen Datenbanken 8, externen Rechnersystemen 9 und/oder Hilfsroutinen 10 verbunden sein.

Bei dem Singletasking-System von Fig. 1 kann in der Regel zu jeder Zeit nur eine Applikation 6 geöffnet sein bzw. arbeiten, weshalb die Applikation 6 alle Ereignisse bzw. alle von dem Preprozessor 7 modifizierten Ereignisse direkt verarbeitet.

In Fig. 2 ist der Ereignisstrom in einer EDV-Anlage mit einem Multitasking-Betriebssystem 5, wie Windows, vereinfacht dargestellt. Bei solchen Multitasking-Systemen können mehrere Applikationen 6 (quasi-)gleichzeitig geöffnet sein und arbeiten. Ein externes Ereignis erzeugt hier ebenfalls einen internen Interrupt, der vom Betriebssystem 5 abgefangen und bearbeitet wird. Im Gegensatz zum obigen Singletasking-System verhält sich das Betriebssystem 5 hier wie ein Verteiler, der die internen Ereignisse jeweils an die entsprechende(n) Applikation(en) 6 verteilt und diese aktiviert. Dazu werden vom Betriebssystem 5 Nachrichten an die Fenster der einzelnen Applikationen 6 gesandt und deren Fensterfunktionen aufgerufen. Die Applikationen 6 erfahren hier eine sogenannte passive Programmkontrolle.

Im Falle des Multitasking-Systems von Fig. 2 ist für jede Applikation 6 ein Preprozessor oder Teil-Preprozessor 7 dem Betriebssystem nachgeschaltet. Die Preprozessoren 7 verarbeiten nur bestimmte Ereignisse, die der entsprechenden Applikation 6 von dem Betriebssystem 5 zugeteilt werden.

Um mit dem Preprozessor 7 in den verschiedenen Systemen wie gewünscht arbeiten zu können, sind mehrere Vorbereitungsschritte nötig, die im folgenden näher beschrieben werden sollen. Im einzelnen sind dies:

V) Nutzungsphase

VI) Funktionalitätsergänzung.

In der Verbindungsphase muß der Preprozessor 7 zunächst physikalisch in das bestehende Rechnersystem eingebunden werden. Hierzu wird bei allen Betriebssystemen 5 ein Ereignistreiber, welcher vorzugsweise ein Bestandteil des Preprozessors 7 sein soll, an eine geeignete Stelle zwischen der Ereignisquelle und der Weiterverarbeitung durch die Applikation(en) 6 eingebunden. Dieser Ereignistreiber muß je nach Art des Betriebssystems 5 die bestmögliche physikalische Einbindung des Preprozessors 7 in den Ereignisstrom bieten. Für Singletasking-Systeme ist beispielsweise ein sogenannter TSR (Terminate Stay Resident)-Treiber verwendbar, welcher die Hardware-Interrupts direkt abfangen kann. Bei Multitasking-Systemen kann der Preprozessor 7 zum Beispiel durch sogenanntes Subclassing der Fensterfunktionen alle Ereignisse vor ihrem Eintritt in die jeweilige Fensterfunktion bearbeiten.

Wenn diese physikalische Verbindung des Preprozessors 7 durch den Ereignistreiber abgeschlossen ist, können sämtliche Ereignisse vor der Bearbeitung durch die Applikation(en) 6 analysiert, gefiltert und evtl. modifiziert werden. Prinzipiell ist nur die Funktion des Ereignistreibers an das jeweilige Betriebssystem 5 anzupassen, alle weiterführenden Analyse- und Synthesefunktionen des Preprozessors 7 beziehen sich ausschließlich auf die abstrakte Darstellung von Ereignissen.

Im zweiten Schritt, der Ereignis-Reaktionen-Analysephase, analysiert der Preprozessor 7 automatisch und/oder manuell die Ereignis-Reaktions-Schnittstellen der jeweiligen Applikation(en) 6. Solche Ereignis-Reaktions-Schnittstellen können zum Beispiel sein: Fokus und Positionen der Eingabefelder in der Bildschirmmaske; Reaktionen auf besondere System-Ereignisse, wie Speichern, Drucken, Ende und dergleichen; unterschiedliche Bedeutung von Ereignissen unter verschiedenen Betriebsbedingungen, wie beispielsweise Eingabe im Wartezustand oder Eingabe während laufender Dateisuche; oder Auswahl verschiedener Betriebsarten der Applikation, wie Maskeneingaben, Suchen, Editieren vorhandener Daten, Kalkulation, Abbuchung, Zubuchung und dergleichen.

Die obige Ereignis-Reaktionen-Analyse kann entweder einmalig vor der eigentlichen Nutzung des erfindungsgemäßen Verfahrens oder als kontinuierlicher Prozeß während des Betriebs der Applikation(en) 6 im Hintergrund erfolgen.

Bei der Ereignis-Reaktionen-Analyse werden laufend Informationen derart gewonnen und verifiziert, daß der Preprozessor 7 ein Ereignis empfängt, dieses unverändert weiterleitet und anschließend die Reaktion der Applikation(en) 6 auf dieses Ereignis bewertet. Solche Reaktionen sind beispielsweise die Anzeige eines Zeichens bei Drücken einer entsprechenden Taste der Tastatur oder das Auslösen der Speicherung auf den Befehl "Speichern". Im Fall von Multitasking-Systemen können durch das oben erwähnte Subclassing alle relevanten Informationen von der Applikation zum Preprozessor umgeleitet werden, da im Betriebssystem intern mit Meldungen desselben gearbeitet wird, die vom Preprozessor relativ leicht zu bewerten sind. Bei Singletasking-Systemen ist die Ereignis-Reaktionen-Analyse schwieriger, es kann aber auch in diesem Fall durch Abfangen aller möglichen Folgeereignisse, wie Cursorpositionen oder Bewertung der Bildschirmanzeige oder Festplattenzugriffe, ein gewisser Rückschluß auf die Reaktionen der Applikation(en) 6 gewonnen werden. Um die Ereignis-Reaktionen-Analyse möglichst vollständig und eindeutig zu machen, ist eine einmalige, manuelle Lernphase des Preprozessors 7 sinnvoll, in der der Benutzer definierte Ereignisse erzeugt und die auf

- I) Verbindungsphase
- II) Ereignis-Reaktionen-Analysephase
- III) Darstellungs-Analysephase
- IV) Aktionenprofilphase

diese Ereignisse entstehenden Reaktionen und deren Bedeutung dem Preprozessor mitteilt.

Nach einer solchen automatischen und/oder manuellen Ereignis-Reaktionen-Analyse sind dem Preprozessor alle relevanten Beziehungen zwischen Ereignissen und Reaktionen von Datenfeldern und Steuerelementen dieser Applikation bekannt.

Es wird nun in einem dritten Schritt, der Darstellungs-Analysephase, die Bedeutung der Ereignisse, die zulässigen Formate der Ereignisse, sowie die Beziehungen zwischen verschiedenen Datenfeldern untersucht.

Diese Darstellungs-Analysephase kann im laufenden Betrieb der Applikation(en) in einem Lernzyklus geschehen, in dem der Preprozessor 7 alle Eingaben in relevante Datenfelder der Applikation überwacht und daraus gewisse Rückschlüsse auf den Zusammenhang zwischen den einzelnen Datenfeldern zieht. Auch in dieser Phase ist eine manuelle Unterstützung durch den Benutzer sinnvoll, der die von ihm gewünschten und zulässigen Datenformate und Datenbeziehungen für die vorhandenen Datenfelder dem Preprozessor beschreibt. In der dieser Analysephase vorangegangenen Ereignis-Reaktionen-Analyse wurden dem Preprozessor die Positionen der Datenfelder vermittelt, und mit dieser Information können nun alle Eingaben zu den Datenfeldern kanalisiert und analysiert werden. Falls möglich, werden den Datenfeldern vom Preprozessor typische Bedeutungen zugewiesen; zum Beispiel ist ein Eingabefeld, das nur Ziffern erhält, ein numerisches Feld, ein Eingabefeld, das nur Vornamen erhält, ein Vornamensfeld, ein Eingabefeld, das nur speziell formatierte Zahlen erhält, ein Datumsfeld, usw.

Wenn die obigen Analysephasen, d. h. die Ereignis-Reaktionen-Analyse und die Darstellungs-Analyse, erfolgreich durchgeführt worden sind, wird anschließend in einem vierten Schritt, der Aktionenprofilphase, festgelegt, mit welchem Aktionenprofil der Preprozessor 7 auf bestimmte erlaubte und unerlaubte Ereignisse oder Verknüpfungen reagieren soll.

Der Preprozessor 7 soll zum Beispiel ungültige Eingabewerte automatisch zurückweisen oder korrigieren, die Darstellung eines Datenwertes verändern (zum Beispiel von "1.1.97" in "01.01.1997"), Datenwerte verändern und filtern, einen oder mehrere Datenwerte umstrukturieren oder eine Fehlermeldung bei unerlaubten Aktionen bzw. Eingaben ausgeben. Ferner kann der Preprozessor 7 Eingaben auch mit externen Datenbanken abgleichen, so daß beispielsweise bei der Eingabe eines Ortes und einer zugehörigen Postleitzahl die Richtigkeit der Postleitzahl anhand der externen Postleitzahldatenbank überprüft und die eingegebene Postleitzahl bei einem Fehler zurückgewiesen oder entsprechend korrigiert (und/oder ergänzt) wird, wodurch die Möglichkeit der Generierung neuer Ereignisse aufgezeigt wird. Desweiteren besteht die Möglichkeit von zusätzlichen Sicherheitsabfragen bei kritischen Aktionen, zusätzlichen Abfragen von Zugriffsberechtigungen, wie Paßworten, und dergleichen.

Zum Teil ist es möglich, grundlegende Aktionenprofile bereits in der Ereignis-Reaktionen-Analysephase und der Darstellungs-Analysephase zu ermitteln. Ferner besteht die Möglichkeit der Einbindung weiterer benutzerspezifischer Aktionenprofile und der Koordination von Aktionen zwischen mehreren gleichzeitig arbeitenden Applikationen, so daß zum Beispiel bei der Neuanlage eines Kunden in der Mailing-Applikation in der parallel arbeitenden Kunden-Applikation nachgefragt wird, ob der Kunde bereits in der Datei vorhanden ist, und bei negativer Auskunft die Neuanlage in der Mailing-Applikation freigegeben wird.

Die Applikationsprofile des Preprozessors 7 können außerdem als Benutzer-Aktionenprofile für jeweils einen be-

stimmten Benutzer gesichert und bei Bedarf wieder geladen werden. Hierdurch kann jeder Benutzer eines Rechnersystems sein persönliches Aktionenprofil für den Preprozessor erstellen und besitzen.

Nach Ablauf der oben beschriebenen, vorbereitenden Phasen sind die Applikationen 6 des Rechnersystems für den Benutzer wie gewohnt nutzbar. Durch den eingestellten Preprozessor 7 werden mittels der gewonnenen Analyse- und Aktionenprofile dabei fehlerhafte Datenerfassungen verhindert, falsche Eingaben unter Umständen aus einem Abgleich mit externen Applikationen oder Datenbanken korrigiert und ergänzt. Auch bei Applikationen, die keine eigene Eingangsprüfung auf fehlerhafte Eingaben aufweisen, ist somit ein Arbeiten mit fehlerbehafteten Daten zuverlässig zu verhindern.

In der eigentlichen Nutzungsphase können ferner bei Bedarf auch weiterhin alle Ereignisse analysiert und die Analyse- und Aktionenprofile gegebenenfalls ständig erweitert, verändert und verifiziert werden.

Sollte dem Benutzer die normale Nutzung der vorhandenen Applikation(en) 6 nicht ausreichen und der Benutzer eine Ergänzung der Funktionen der Applikation(en) wünschen, so ist dies innerhalb eines gewissen Rahmens durch eine spezielle Programmierung des Preprozessors 7 ebenfalls möglich. Hierdurch wird es dem Benutzer ermöglicht, weiterhin mit der ihm vertrauten Applikation zu arbeiten und zusätzliche Funktionen in diese Applikation einzubinden, die ihm bisher gefehlt haben.

So können beispielsweise Funktionen aus der Applikation entnommen, mit dem Preprozessor bearbeitet und in überarbeiteter Form wieder der Applikation bereitgestellt werden. Denkbar ist so zum Beispiel, daß der Preprozessor im Falle einer unübersichtlichen Eingabemaske der Applikation ein neues Fenster mit Datenfeldern erzeugen kann, in dem die gesamte Funktionalität einer vom Benutzer gewünschten Eingabemaske programmierbar ist. Nach Änderung der Maske werden die virtuellen Datenfelder in die entsprechenden Datenfelder der Applikation übertragen, der Benutzer arbeitet dann nicht mehr mit der Original-Eingabemaske der Applikation, sondern mit der von ihm modifizierten Eingabemaske.

Weiter kann der Preprozessor sogenannte intelligente Eingabefelder erzeugen, indem der Preprozessor die Eingabefelder während der Eingabe überwacht und, falls falsche Eingaben in ein Datenfeld eingegeben wurden, die Daten automatisch in das richtige Eingabefeld kopieren.

Ebenso können die Applikationen um fehlende Funktionen ergänzt werden, wie zum Beispiel um eine statistische Auswertung.

Hierzu vereinbart der Preprozessor ein neues Ereignis, wie "Zeige Statistik", mit dem ein neues Fenster geöffnet werden kann, in welchem der Benutzer zuvor die von ihm gewünschte neue Funktionalität, d. h. beispielsweise das Anzeigen von statistischen Werten, einprogrammiert hat. Der Preprozessor kommuniziert hierbei mit der Applikation im Hintergrund und stellt die gewünschten Daten bereit. Für den Benutzer wird diese neue Funktion quasi ein Bestandteil der Applikation.

Für diese Funktionen kann der Preprozessor auf weitere externe Applikationen zugreifen, wie zum Beispiel auf eine Tabellenkalkulation, auf Grafikprogramme, Statistikprogramme usw., und arbeitet zwischen der aktiven Applikation und den möglichen externen Applikationen als Bindeglied, Steuerelement oder Manager.

Eine weitere Möglichkeit des Preprozessors besteht in der Schaffung von Online-Funktionen. So kann der Preprozessor Ereignisse auch auf mehrere Applikationen in der Art verteilen, daß der Preprozessor Verbindungen, sogenannte

Links, zwischen bestimmten Datenfeldern erzeugt und verwaltet. Beispielsweise kann ein Eingabefeld in US\$ mit einem Link zu einem Finanzprogramm die Umrechnung von US\$ in DM mit dem momentan aktuellen Börsenkurs durchführen. Diese Verbindungen können auch über interne und externe Netzwerke bestehen, um einzelne Applikationen mit einem größeren Rechnerverbund zu vernetzen. Dabei können Teile der Funktionen der Applikation von dem Preprozessor und/oder der über einen externen Link verbundenen zweiten Applikation übernommen werden, wobei der nötige Programmablauf im Preprozessor durch eine einfache Programmiersprache vom Benutzer programmiert werden kann.

Durch die freie Programmier- und Konfigurierbarkeit des Preprozessors können auf einfache Weise bestehende Applikationen nahezu beliebig verändert und auf die Bedürfnisse des jeweiligen Benutzers abgestimmt werden.

Aus einem Pool von verschiedenen Applikationen kann der Benutzer seine optimale Konstellation zusammenstellen.

Sämtliche in den Figuren gezeigten und in der Beschreibung erläuterten Einzelheiten sind für die Erfindung wichtig. Dieses gilt gleichermaßen auch für die Zusammenfassung, die auch zur Auslegung des Schutzzumfangs herangezogen werden kann.

Patentansprüche

1. Elektronisches Datenerfassungsverfahren, mit den Verfahrensschritten Erfassen von Eingaben und Weiterleiten der Eingaben an eine Applikation des Rechnersystems, dadurch gekennzeichnet, daß die Eingaben vor der Weiterleitung an die Applikation durch einen Preprozessor (7) auf ihre Richtigkeit überprüft und fehlerhafte Eingaben zurückgewiesen oder korrigiert werden.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für ein Singletasking-Betriebssystem die Überprüfung und Verarbeitung der Eingaben vor ihrer Weiterleitung an das Betriebssystem (5) erfolgt.
3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß für ein Multitasking-Betriebssystem die Überprüfung und Verarbeitung der Eingaben nach der Verteilung durch das Betriebssystem (5) und vor ihrer Weiterleitung an eine der Applikationen (6) erfolgt.
4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß vor der Nutzung des Preprozessors (7) bzw. des Datenerfassungsverfahrens die Eingaben automatisch und/oder manuell mit Unterstützung durch den Benutzer analysiert werden.
5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß die Analyse der Eingaben in zwei Stufen erfolgt, wobei in der ersten Stufe die Reaktionen der Applikation(en) (6) auf Eingaben und in der zweiten Stufe Bedeutung und Zusammenhänge von Datenfeldern analysiert werden, und daß aus der Analyse der Reaktionen auf Eingaben ein Analyse- und Aktionsprofil erzeugt wird, anhand dessen die Überprüfung und Verarbeitung der Eingabe erfolgt.
6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, daß für jeden Benutzer ein benutzerspezifisches Analyse- und Aktionsprofil erstellt wird.
7. Verfahren nach einem der vorgenannten Ansprüche, dadurch gekennzeichnet, daß der Preprozessor (7) vorbereitend für die Nutzung zunächst in einer Verbindungsphase physikalisch in das Rechnersystem eingebunden wird, in einer Ereignis-Reaktionen-Analysephase die Ereignis-Reaktionsschnittstellen der Appli-

kation analysiert, in einer Darstellungsanalysephase die Ereignisse und Datenfelder untersucht und eine Aktionenprofilphase durchläuft zur Bestimmung der Reaktionen auf erlaubte und unerlaubte Ereignisse oder Verknüpfungen sowie zur Ermittlung der speziellen Verknüpfungsstrukturen der zuvor bestimmten Elemente, und daß anschließend der Preprozessor (7) in einer Nutzungsphase Ereignisse analysiert, Aktionen- und Analyseprofile erweitert, verändert und verifiziert und eine Funktionalitätsergänzung erlaubt.

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß die Applikationen (6) um neue Funktionen erweitert werden, die als virtueller Bestandteil der Applikationen nutzbar sind.

9. Vorrichtung zur Durchführung eines elektronischen Datenverarbeitungsverfahrens mit einer Eingabeschnittstelle für die Eingabe von Daten, mindestens einer Applikation (6) und einem Betriebssystem (5), insbesondere zur Durchführung des Verfahrens nach einem der vorhergehenden Ansprüche, gekennzeichnet durch mindestens einen Preprozessor (7) zur Überprüfung der Eingaben auf ihre Richtigkeit, wobei der Preprozessor geeignet ist, fehlerhafte Eingaben zurückzuweisen oder zu korrigieren.

10. Vorrichtung nach Anspruch 9, gekennzeichnet durch einen Ereignistreiber zur physikalischen Einbindung des Preprozessors (7) in den Ereignisstrom.

11. System nach Anspruch 9, dadurch gekennzeichnet, daß der Ereignistreiber ein Bestandteil des Preprozessors (7) ist.

Hierzu 2 Seite(n) Zeichnungen

- Leerseite -

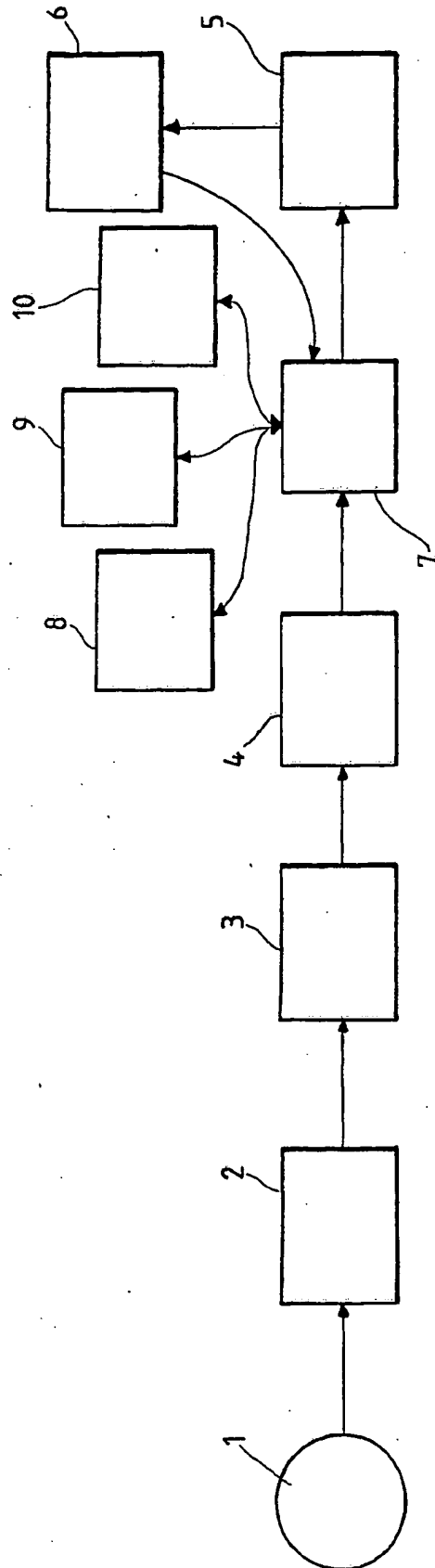


FIG. 1

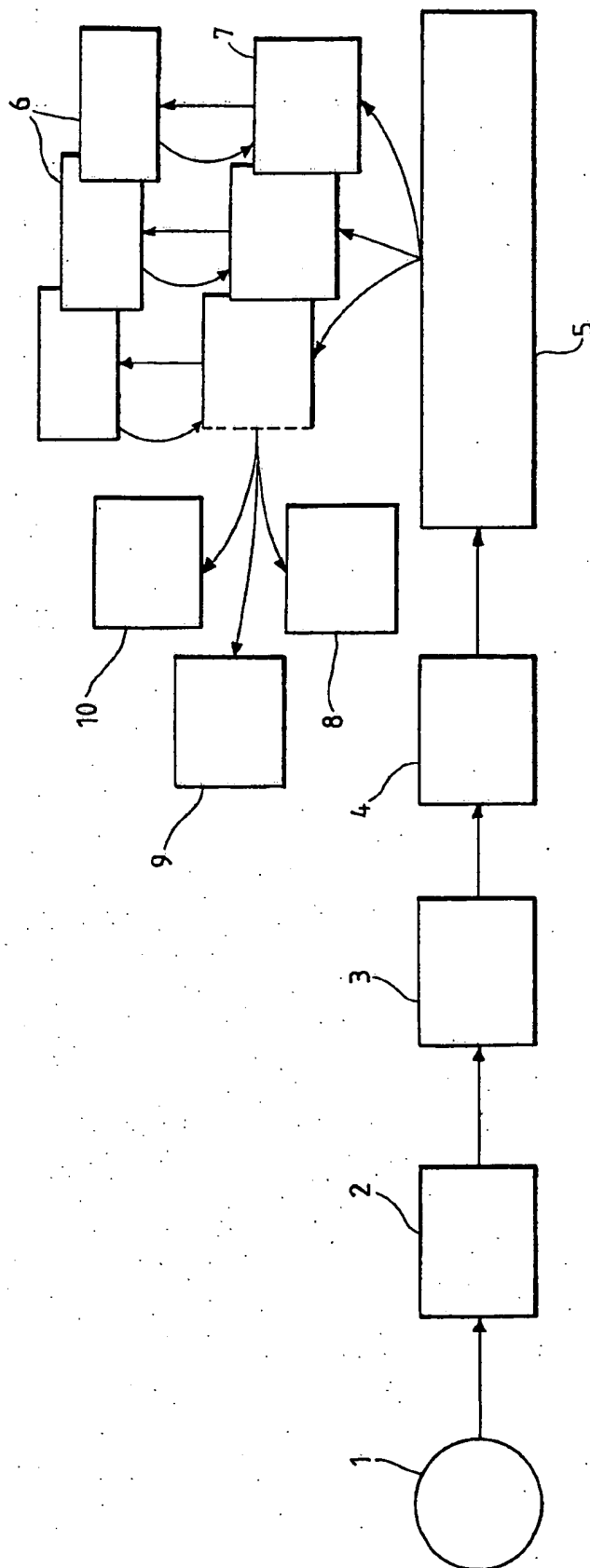


FIG. 2